Тестовое задание

Часть 1

таблицы:

users
---
id (pk)
email (nullable)

subscriptions
---
id (pk)
user_id (fk users.id)
status (active | canceled | expired)
plan_id
current_period_start
current_period_end

payments
---
id (pk)
external_payment_id
provider (stripe, yookassa, etc)
user_id (nullable)
subscription_id (nullable)
amount
currency
status (pending | succeeded | failed)
paid_at
raw_payload

webhook_events
---
id (pk)
provider
external_event_id
event_type
status (received | processed | failed)
received_at
processed_at
payload
error

Уникальность:

users: email - уникальность пользователя
subscriptions: user_id - статус подписки может быть только один для пользователя
payments: связка provider и external_payment_id - payment может быть только один, нужна связка для избежания коллизий разных провайдеров
webhook_events: связка provider и external_event_id - event может быть только один, нужна связка для избежания коллизий разных провайдеров

Индексы:
users: email - быстрый поиск пользователя
subscriptions: user_id - поиск по пользователю, status, current_period_end - так как нужны для поиска и проверки статуса
payments: provider, external_payment_id - нужны в дальнейшем при поиске для обработки дублей, а также для поиска: user_id, status, subscription_id
webhook_events: связка provider и external_event_id - status - для поиска и отслеживания

Часть 2:

handleWebhook(request):

```
handleWebhook(request):

if !verifySignature(request):
    return 401

payload = parse(request)
require payload.external_event_id, payload.payment_id


event = getOrCreateWebhookEvent(event_id)

if event.status == 'processed':
  return 200


begin transaction
try:

    payment = getOrCreatePayment(
        provider = payload.provider,
        external_payment_id = payload.payment_id,
```

```
    paid_at = payload.paid_at,
    amount = payload.amount
)


if payment.applied:
    markWebhookProcessed()
    commit
    return 200


if payload.payment_status != 'succeeded':
    payment.markFailed()
    markWebhookProcessed()
    commit
    return 200


if !amountMatchesPlan(payment, payload.plan_id):
    payment.markFailed("amount mismatch")
    alert()
    markWebhookProcessed()
    commit
    return 200


user = resolveUser(
    payment.user_id,
    payload.external_customer_id,
    payload.email
)

payment.bindUser(user)


subscription = lockSubscription(user)

base_time = max(
    subscription.current_period_end,
    payment.paid_at
)

subscription.extendFrom(base_time)

payment.markSucceeded(subscription)
markWebhookProcessed()

commit
```

```
    return 200

  except Exception e:
    rollback
    markWebhookFailed(e)
    return 500
```

Часть 3.

1. webhook пришел дважды
уникальный (provider, external_event_id)
сразу 200, без побочных эффектов

2. webhook пришел раньше создания user
user.email nullable и создаем user без email
позже можно обработать

3. webhook пришел без email, но есть externalPaymentId
обрабатываем через payments.external_payment_id, user создается без email
подписка активируется

4. webhook пришел с другой суммой, чем план
логируем, не активируем подписку автоматически
статус failed или условный pending, алерт в мониторинг

5. webhook пришел через неделю
используем paid_at, а не received_at, корректно считаем период подписки, продление от current_period_end, а не от now

6. сервер упал после записи payment, но до subscription
payment уже сохранен
webhook_event = received

при повторе webhook:
payment найден
subscription будет создан

Часть 4.

Логи:

На каждый webhook:
provider
external_event_id
external_payment_id

payment_status
amount / currency
webhook_event.status
payment.id / subscription.id
error (если есть)

Алерты:
webhook пришел с другой суммой, чем план
webhook failed
payment succeeded, subscription not updatet